

# Conception et architectures des réseaux

## Exercices pratiques

Dernière modification: 27/4/2020

### Exercice 3 Transfert de fichier par TCP

Vous voulez transférer un fichier très volumineux (d'une taille de  $L$  octets) de A vers B sur un réseau Ethernet, avec un MSS de 1460 octets.

1. Quelle est la valeur maximale de  $L$  compatible avec les numéros de séquence de TCP? Souvenez-vous que le champ de numéros de séquence TCP dispose de 4 octets.

La valeur maximale de  $L$  compatible avec les numéros de séquence de TCP est la taille maximale avec laquelle les numéros de séquence des segments de données de ce fichier ne bouclent pas. C'est à dire au-delà de cette longueur, on trouvera des numéros de séquence qui ont été utilisés précédemment. Remarquons que selon le mécanisme de calcul de numéro de séquence de TCP, le numéro de séquence du segment suivant n'est pas incrémenté une unité mais égal au numéro de séquence du segment courant plus sa longueur, c-à-d qu'il y a un saut de numéro. Avec le transfert d'un fichier long, les numéros de séquence constitueront une suite arithmétique de raison de MSS. Comme le compteur de numéro de séquence est un entier non signé de longueur fixe (32 bits dans le cas de TCP), alors il va tourner en rond lorsqu'on dépasse sa limite (le numéro de séquence maximal autorisé,  $2^{32} - 1$  dans notre cas). Soient  $n$  le nombre de numéros de séquence disponibles ( $2^{32}$  dans notre cas) et  $m$  le MSS (1460 dans notre cas),  $n > m$ , alors on a :

- a) La borne supérieure  $L_{MAX} = n * m$  (tous les numéros de 0 à  $n-1$  sont utilisés)
- b) Si  $m$  divise  $n$ , soit  $q = n/m$ , on voit facilement qu'après  $q$  segments, le numéro de séquence commence à boucler et dans ce cas,  $L_{max} = q * m$  (octets).
- c) On peut démontrer que si  $n$  et  $m$  soient premiers entre eux (1 est leur seul diviseur commun), on atteindra la borne supérieure  $n * m$  ci-dessus, c-à-d les tours en rond des numéros de séquence vont combler au fur et à mesure les « trous » des numéros sautés après l'envoi de chaque segment.
- d) On en déduit facilement que si  $n$  et  $m$  ont des diviseurs communs et  $\{g = PGCD(n, m) \mid n = r * g \wedge m = p * g \wedge (n \& p \text{ premiers entre eux})\}$ , alors  $L_{max} = r * g * p = n * p = r * m$  (octets).

Dans notre cas,  $n = 2^{32}$  et  $m = 1460$ , on trouve que  $g = PGCD(2^{32}, 1460) = 4$ ,  $r = n/g = 2^{30}$   
 $L_{max} = r * m = 2^{30} * 1460 = 1,567663063 \times 10^{12} = 1\ 567\ 663\ 063\ 000$  (octets) = 1 460 GO  
Le nombre de segments est  $r = 2^{30}$

2. Partant de la valeur  $L$  obtenue ci-dessus, calculez le temps nécessaire à la transmission intégrale du fichier. Nous admettrons ici qu'une longueur d'en-tête totale (en-têtes de transport, réseau et liaison de données confondus) de 66 octets est ajoutée aux différents segments avant que le paquet final ne soit constitué et envoyé sur une liaison à 10 Mbit/s. Ignorez les contrôles de flux et de congestion. A est ici capable d'envoyer ses segments les uns après les autres en continu.

La taille du trame (de bits) correspondant à un segment TCP est :

$$S_{trame} = (1460 + 66) * 8 = 12208 \text{ (bits)}$$

Le temps nécessaire pour la transmission d'un segment de données TCP sur un réseau 10Mbps est

$$T_{trame} = S_{trame} / (10 * 2^{20}) = 0,001164246 \text{ (s)}$$

Le temps nécessaire pour la transmission intégrale du fichier ( $r$  segments) sera alors :

$$T_{fic} = T_{trame} * r = 0,001164246 * 2^{30} = 1250099,623624704 \sim 1250100 \text{ (s)} = 347,25 \text{ (h)} = 347:15 = 14j11h15'$$

3. Qu'est-ce que change si vous utilisez un réseau moins fiable (par exemple: wifi) avec un MSS de 536 octets?

En remplaçant la valeur de  $m$  par 536, nous aurons :

$$g = PGCD(2^{32}, 536) = 8, r = n/g = 2^{29}$$

$$L_{max} = r * m = 2^{29} * 536 = 287\,762\,808\,832 \text{ (octets)} = 268 \text{ (GO)}$$

$$S_{trame} = (536 + 66) * 8 = 4816 \text{ (bits)}$$

$$T_{trame} = S_{trame} / (10 * 2^{20}) = 0,00045929 \text{ (s)}$$

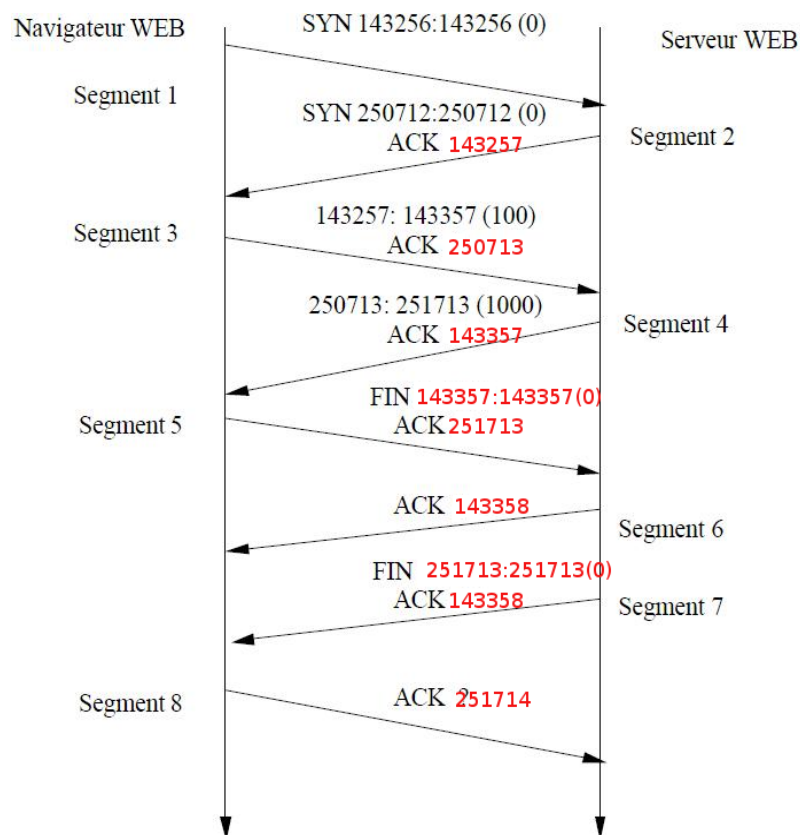
$$T_{fic} = T_{trame} * r = 0,00045929 * 2^{29} = 246579,44117248 \text{ (s)} \sim 68,5 \text{ (h)} = 68:30 = 2j20h30'$$

#### Exercice 4 Échange TCP

L'échange TCP de la figure ci-dessous correspond au transfert d'une page WEB entre un navigateur WEB et un serveur WEB. On fait l'hypothèse que la requête à la page WEB fait 100 octets et que la page WEB retournée fait 1000 octets. Il n'y a pas d'erreurs de transmission.

Pour chaque segment de données, différentes informations apparaissent. D'une part la présence d'un ou plusieurs des différents indicateurs comme SYN, FIN, ACK. Par ailleurs sur la première ligne deux chiffres sont portés. Le premier chiffre correspond au numéro de séquence du premier octet du segment, le deuxième chiffre correspond au numéro du premier octet du prochain segment à envoyer. Le chiffre entre parenthèses correspond au nombre total d'octets transmis dans le segment. Si le segment est porteur d'un acquittement positif, l'indicateur ACK est mentionné et a coté de lui doit figurer la valeur du champ acquittement du segment TCP.

Complétez les numéros de séquence et les numéros d'acquitements qui manquent sur la figure (qui apparaissent sous forme de point d'interrogation). Indiquez à quoi correspondent les différents segments numérotés de 1 à 8. Inspirez-vous des résultats du premier TP sur l'étude de TCP (analysez les captures des trames pour voir comment évoluent les numéros de séquence et des acquitements, les fanions, etc.)



#### Exercice 5 Transmission de fichier par TCP avec fenêtres statiques

Imaginez la transmission d'un fichier  $O$  de 100 Koctets d'un serveur vers un client. Soit  $S = 536$  octets et  $RTT = 100 \text{ ms}$ . Admettons que le protocole de transport utilise des fenêtres statiques, de taille  $W$ . Pour un débit  $R$  de 28 kbit/s, déterminez le temps de latence minimum possible. Déterminez par ailleurs la taille de la fenêtre minimale compatible avec ce temps de latence.

On aura la latence minimum si les segments du fichier soient transférés continuellement. Autrement dit la taille du fenetre doit etre superieure ou egale à la quantite de donnees (en segments comme unite) que la source peut transmettre en attendant un acquittement (1 RTT). Ainsi, le rythme de l'envoi des segments et celui des acquittements sont égaux et la transmission est sans interruption. On y ajoutera un RTT a la fin pour être informer que le fichier a été reçu entièrement (TCP oblige). Bien entendu, la taille de la fenetre doit  $\geq 2$  dans tous les cas.

Selon les calculs de l'exercice 3, on a :

$$n = O/S = (100 \cdot 2^{10}) / 536 = 192 \text{ (segments)}$$

$$S_{trame} = (536 + 66) \cdot 8 = 4816 \text{ (bits)}$$

$$T_{trame} = S_{trame} / R = 4816 / (28 \cdot 2^{10}) = 0,16796875 \text{ (s)}$$

On constate qu'avec ce réseau de débit  $R$  de 28 kbit/s, le temps de transmission d'un trame (0,16796875 (s)) dépasse largement le RTT (0,1 s). Par conséquent, il suffit d'une fenetre de taille minimum  $W = 2$  trames =  $536 \cdot 2 = 1072$  octets, nous pouvons déjà transférer continuellement le fichier. Et la latence minimum pour le transfert de ce fichier sera:

$$T_{fic} = T_{trame} \cdot n + RTT = 0,16796875 \cdot 192 + 0,1 = 32,25 + 0,1 = 32,35 \text{ (s)}$$

Même question pour les débits suivants:  $R = 100 \text{ kbit/s}$ ;  $1 \text{ Mbit/s}$ ;  $10 \text{ Mbit/s}$ ;  $100 \text{ Mbit/s}$ .

$$\mathbf{R=100 Kbs} : T_{trame} = S_{trame} / R = 4816 / (100 \cdot 2^{10}) = 0,04703125 \text{ (s)}$$

Au début, il faut donc envoyer 3 segments de suite pour qu'on reçoive l'acquittement du 1<sup>er</sup> segment et au-delà, le rythme d'arrivée des acquittements sera soutenu. Par conséquent,

$$W_{min} = \sup(0,1 / 0,04703125) = 3 \text{ (segments)}$$

$$\mathbf{R=1Mbs} : T_{trame} = S_{trame} / R = 4816 / (2^{20}) = 0,004592896 \text{ (s)}$$

$$\text{Même raisonnement, la taille de fenetre, } W_{min} = \sup(0,1 / 0,004592896) = 22 \text{ (segments)}$$

...

#### Exercice 6 Transmission de fichier par TCP avec fenêtres dynamiques

Vous transférez un fichier de taille  $O = 100 \text{ Koctets}$  en utilisant une connexion TCP avec  $RTT = 1 \text{ seconde}$  et  $S = 536 \text{ octets}$ . Comparez le temps de latence minimum avec le temps de latence observé en présence d'une phase de départ lent, pour les débits suivants :

$R = 28 \text{ kbit/s}$ ,  $100 \text{ kbit/s}$ ,  $1 \text{ Mbit/s}$ ,  $10 \text{ Mbit/s}$  et  $100 \text{ Mbit/s}$ .

Donnez vos conclusions.

Selon les calculs de la question 5 avec le  $\mathbf{R=28Kbs}$  et  $RTT=1s$ , la taille de fenetre  $W_{min}$  pour avoir le latence minimum est

$$W_{min} = \sup(1 / 0,16796875) = 6 \text{ (segments)}$$

alors le temps de latence observé en présence d'une phase de départ lent est le suivant :

$$T(1^{er} \text{ segment}) = T_{trame} + RTT = 0,16796875 + 1 = 1,16796875, W = 2$$

$$T(2^{e} + 3^{e}) = 2 \cdot T_{trame} + RTT - T_{trame} = 0,16796875 + 1 = 1,16796875, W = 4$$

$$T(4^{e} + 5^{e} + 6^{e} + 7^{e}) = 4 \cdot T_{trame} + RTT - 3 \cdot T_{trame} = 0,16796875 + 1 = 1,16796875, W = 8$$

A partir de cette taille, pour les segments restants ( $192 - 7 = 185$ ) prennent le latence minimum qui est

$$T_{rest} = T_{trame} \cdot n + RTT = 0,16796875 \cdot 185 + 1 \sim 32 \text{ (s)}$$

$$T_{file} = 3 \cdot T(1^{er}) + T_{rest} = 3 \cdot 1,16796875 + 32,07421875 = 35,578125 \text{ (s)}$$

Même raisonnement pour  $\mathbf{R = 100Kbps}$ , ....